Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations	1. Thesis and Dissertation Collection, all items

1988-03

Ada implementation issues as discovered through a literature survey of applications outside the United States

Soong, Warren J.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/23518

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDL[illegible]
NAVAL P[illegible]
MONTEREY, CALIFORNIA 93943-5002

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b. OFFICE SYMBOL *(If applicable)*
37

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

Program Element No	Project No	Task No	Work Unit Accession Number

11. TITLE *(Include Security Classification)*
ADA IMPLEMENTATION ISSUES AS DISCOVERED THROUGH A LITERATURE SURVEY OF APPLICATIONS OUTSIDE THE UNITED STATES

12. PERSONAL AUTHOR(S) Soong, Warren J.

13a. TYPE OF REPORT
Master's Thesis

13b. TIME COVERED
From To

14. DATE OF REPORT *(year, month, day)*
March 1992

15. PAGE COUNT
89

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
Ada

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

The Ada programming language has been adopted, mandated and legislated for use by the Department of Defense for all software development where cost effective. Although it is a powerful, general purpose language, Ada has encountered some resistance to its implementation within DoD. However, Ada's use outside DoD, as well as outside the United States, is on the rise. Foreign governments and commercial industries are realizing Ada's benefits throughout the software development cycle. The acknowledgment of Ada's advantages in productivity enhancement and cost saving has initiated a shift towards Ada, away from older programming languages, like Fortran and COBOL. This study aims to disseminate information about the language and its use around the world, as well as discuss negative attitudes which present the biggest challenge to Ada's implementation.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS REPORT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Martin J. McCaffrey

22b. TELEPHONE *(Include Area code)*
(408) 646-2488

22c. OFFICE SYMBOL
AS/MF

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

T258584

Approved for public release; distribution is unlimited.

Ada Implementation Issues As Discovered Through A Literature Survey Of Applications Outside The United States

by

Warren J. Soong
Captain, United States Marine Corps
B.A., Cornell University, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

The Ada programming language has been adopted, mandated and legislated for use by the Department of Defense for all software development where cost effective. Although it is a powerful, general purpose language, Ada has encountered some resistance to its implementation within DoD. However, Ada's use outside DoD, as well as outside the United States, is on the rise. Foreign governments and commercial industries are realizing Ada's benefits throughout the software development cycle. The acknowledgment of Ada's advantages in productivity enhancement and cost saving has initiated a shift towards Ada, away from older programming languages, like Fortran and COBOL. This study aims to disseminate information about the language and its use around the world, as well as discuss negative attitudes which present the biggest challenge to Ada's implementation.

3665765
c.1

TABLE OF CONTENTS

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

I. INTRODUCTION

A. BACKGROUND

Ada was created to be the single programming language for the Department of Defense (DoD) to replace about 450 other computer languages being used throughout the services. Ada was developed to be a general purpose language suitable for a wide variety of software applications - from embedded weapons systems to command and control to management information systems. It can be used in the place of other high-level programming languages in all areas, not just for defense-oriented purposes. This research aims to investigate the use of Ada outside the defense market. It will focus on applications outside the United States and how lessons learned from those markets can be applied to DoD implementation of Ada.

Outside of the United States Ada is used primarily in ministries of defense and defense industries. Its use is also expanding within governmental organizations not related to defense. The goal of this study was to discover whether Ada plays a significant role in systems software development outside the United States. In addition, how Ada may have achieved greater acceptance overseas, due to cultural and managerial differences, is examined.

B. RESEARCH QUESTIONS

The original focus of this research was about how extensively Ada is used in commercial applications outside the United States. However, the data preponderance was predisposed toward a discussion of why Ada was, and was not, being employed for systems development when compared with other options.

The primary research question is why were organizations, especially commercial organizations, outside DoD adopting Ada for many types of applications when many DoD development activities seemed to be resisting implementation of Ada in their software? This question is addressed along managerial and attitudinal lines.

Secondary questions include:

- How rapidly is Ada usage increasing outside DoD sponsorship?
- For what types of projects is Ada being used?
- What are the most important factors for choosing Ada?
- What kind of results are being achieved with Ada?
- What other options were considered?

Language implementation issues and results of using Ada versus other programming languages comprised the bulk of the information gathered. Types of applications and organizations using Ada were also investigated.

C. METHODOLOGY

Data was gathered through a comprehensive literature search of industry publications as well as material from Ada vendors and government sources. The key sources discovered during the literature review are listed in the reference section of this study. It was found that there are also several independent surveys of the commercial Ada market, available at significant cost. (Focused Ada Research, 1991; Software Strategy & Tactics, 1991) They divide Ada usage into specific geographic areas as well as into application types. Truly comprehensive studies of corporate use of Ada are extremely time and money consuming.

The researcher attended the 1991 Ada-Europe Conference and the 1991 Tri-Ada Conference provided information on current work in the Ada market. The conferences also provided an opportunity to establish contact with organizations and individuals within the Ada field. Time, distance and resource constraints prohibited a closer, first-hand view of Ada usage outside the United States.

D. OVERVIEW

Chapter II provides an overview of the Ada programming language, reviewing its history and discussing the advantages of the language over other high-level languages. The advantages covered were given by users of Ada as reasons for choosing the language over other options.

As Ada develops and its market diversifies, changes in the environment enhance Ada usage. However, there remain conditions, primarily attitudes, which block the acceptance of Ada as a suitable option for software development. Chapter III discusses these issues that are involved in Ada's adoption and implementation. Reasons for the increasing popularity of Ada, as well as barriers to Ada's adoption, are examined.

Chapter IV surveys the use of Ada outside the United States in Europe, Australia and Japan. There is documented use of Ada throughout the world, on all seven continents. Additional information, on specific applications, is found in the Appendix.

Ada has a sufficiently stable foundation, both in defense and commercial industry, to allow its growth and establishment within the programming market. Chapter V presents some thoughts about what the future holds for Ada, and DoD's role in that future.

II. ADA PROGRAMMING LANGUAGE

This chapter provides a brief historical perspective on Ada and a discussion of the language's major advantages. Specific examples are provided.

A. HISTORY

The rapid technological advances in computer hardware development led to the belief that software advances would be similarly possible and at similar cost. Because software is difficult to specify, write, debug and maintain, and given the demand for software, a software crisis occurred. Because of poor project management and individualistic programming styles, the software crisis led to the concept of structured programming in which rules and recommendations for good programming were followed. Structured programming requires structured languages.

As sizes of projects became increasingly larger and more complex, with larger teams of programmers, even well-structured programs were not sufficient to deal with requirements.

In the early 1970's, the U.S. Department of Defense (DoD) realized it was spending a significant amount of money on software, particularly in embedded systems. A major cost driver was that DoD systems were written in over 450 languages, making development and maintenance of the programs even more expensive. (Long, 1991, p. 5) Based upon previous demand for software, outyear growth projections were unaffordable.

In an attempt to reduce the costs associated with the multiplicity of languages, DoD decided to standardize on one language. In 1975, they established a high order language working group to develop specification requirements which were continually refined. The final set of requirements resulted from the collective evaluation of more than 80 review teams. Instead of using a base language and appointing an implementation team, the working group opted for an open design competition.

A request for proposal for language development was issued in May 1977. Seventeen language designers submitted proposals. Of these, four were chosen for a six-month preliminary design. Each vendor had chosen Pascal as the base language. After extensive review, two contractors were chosen for a year-long development. The two designs were delivered in March 1979. They were subjected to intensive analysis by more than 50 teams of both government and industry reviewers. The European team from Honeywell Bull in Paris won the competition in 1979.

The language was named Ada after the Countess Augusta Ada Lovelace, generally considered the world's first computer programmer. After extensive evaluation and testing, minor changes were made and Ada was published as MIL-STD 1815 in 1980. In 1983, Ada was declared an American National Standards Institute (ANSI) standard and, in 1987, an International Standards Organization (ISO) standard. (Skansholm, 1988)

The hope was that Ada would be the programming language to remedy DoD's software crisis. It would provide modular development, structure, standardization and flexibility. In addition, the facilities to support programs in a variety of systems would be furnished. The development resulted in a general purpose, standardized high-order language suitable for most applications.

B. LANGUAGE ADVANTAGES

It is Ada's software engineering attributes, reusability and portability which make it an appealing language, both for defense markets and commercial industry. Ada, like Pascal, possesses well-designed control and data structures that make it suitable for a wide range of applications. In addition, Ada supports concurrent tasking and real-time programming, features not present in most high-level languages. Ada was designed, primarily, to promote adherence to modern software engineering principles. This results in software which is reliable, maintainable, portable, readable and reusable.

1. Modularity

Ada is a well-structured language containing control features which make it easier to understand and more readable. Although structured features are not unique to Ada (high level control statements, program and data type facilities are also supported by Pascal) it greatly enhances these features, mainly to support the concept of modular development.

The language features of Ada encourage programmers to develop code which encapsulates objects and their related operations into a single program unit, which can be planned, written, compiled and tested alone. Ada provides a means of creating abstract objects - similar to objects found in the real world - allowing the programmer to specify and compile these as interfaces. These interfaces remain fixed and separate from their implementation. This is referred to as information hiding - separating the how from the what - the "black box" principle. It promotes independence of modules, allowing changes which will not affect other parts of the program.

2. Error Handling

Ada provides a means of handling and recovering from run-time errors. Ada's exception handler allows a system to continue to function despite the presence of an undefined or unmeasurable value. The error value will not cause a system failure, but merely a degradation of service because the error can be handled - identified and terminated - prior to causing damage. This prospect of fault tolerance gives Ada an advantage over other languages for critical systems.

For software design error avoidance Ada has been designed for reliability. The language itself encourages good programming style and reuse of proven components (packages).

3. Parallel Processing

Ada tasks are entities whose executions proceed in parallel in the sense that each can be considered to be executed by a logical processor of its own. Different tasks proceed independently, except at points where they synchronize. Tasks can have entries by which that task is called by other tasks. Parallel tasks may be implemented on multiprocessors or with interleaved execution on a single physical processor. (ANSI/MIL-STD 1815A - 1983) Parallel processing and tasking are key Ada features for real-time applications.

The air traffic control systems in Europe and the U.S. have chosen the Ada language for their systems. They depend heavily on real-time processing and are two examples where Ada's language defined tasking is an obvious advantage. Another company, Euristic Systems, chose Ada for developing an expert system which incorporates temporal reasoning to manage the continuous acquisition of data because of its real-time primitives. (Brosgol, 1990)

4. Reusability

There are several features which promote code reuse. The packages of Ada are collections of sub-programs, data types, data objects, procedures, functions and other constructs. They are prefabricated components usable by any program. This leaves the development of complex inner workings of programs to experts in the field, allowing less experienced programmers to build upon the foundational labor of others.

It is through packages that Ada provides the means to define abstract data types. This makes it possible to extend the language. New packages would constitute a specific extension, replacing standard definitions with specialized versions. It is possible to build up a library of prefabricated packages which could be used in various contexts within different programs.

The generic is another unique structure in Ada which enhances reusability. Generics are a way of writing templates of procedures, functions or packages which can be implemented differently depending upon the specific process. There always exists a specification and a body. However, a generic cannot be called unless an actual parameter is supplied. Thus, there can be a parameterized package which describes the shape of procedures and is instantiated as a new package with specified types.

Reusability demonstrates its advantages in a reduction of costs through reduced code developments as well as an increase in reliability of software.

System Automation Corp, a U.S. corporation which implemented a recruiting and retention system for the U.S. Army, gave the potential reuse of components as a reason for choosing Ada. (Brosgol, 1990) Reuse is a clear result of choosing Ada for project development due to the nature of the language. Ada seeks the separation of logical operations from their physical implementation. This creates separate modules which are capable of performing logical functions that can be used anywhere. Thus, reusability is achieved (via Ada's packages, tasks and generics) more easily than in other programming languages.

5. Portability

Because of the standardization of the Ada language and the rigorous validation procedures undertaken for compilers, Ada achieves a level of portability among systems not approached by other programming languages.

As software systems become more complex they are no longer able to function independently. Systems interface with external I/O devices or communications equipment. Typically the design requires device independence.

The portability advantage of language standard-ization and compiler compatibility was a factor for Thomson-CSF, a French company, in their development of air traffic control systems. (Brosgol, 1990) This was also the case for Volvo, the Swedish car manufacturer, for their parts delivery system at the assembly site. (Crafts, June 1990, pp. 3-8)

The maturity of compilers, achieved through vigorous validation, is vital in attaining portability across systems and was a reason cited by a United States Marine Corps logistics base which chose Ada for a management information system application. (Crafts, May 1989, pp. 1-3) The resulting portability across development operating systems and real-time executives was a reason Euristic Systems, a French engineering company, used Ada to implement real-time products in artificial intelligence. (Brosgol, 1990) Lastly, Motorola

Radio-Telephone Systems Group, a U.S. company which developed an automated software tool in Ada to perform regression testing on cellular telephone switches, wanted easier system integration which they were unable to obtain from other programming languages. (Crafts, July 1989, pp. 8-12)

6. Maintainability

Much of the maintainability of Ada programs is due to their readability and ease of documentation. Modularity and information hiding add to the level of maintainability, along with separate specification and compilation of units. The strict adherence to correct software engineering procedures stressed in Ada programming leads to better written code with fewer bugs in the system.

Organizations, ranging from dSPACE Digital Signal Processing (a German company which deals in tools to control fast systems, such as Winchester drives and vehicle suspension systems) to Ecole Normale Superieure (a French graduate school in mathematical and biological sciences), have considered Ada's maintenance benefits as key advantages when selecting it as the programming language for their systems. (Brosgol, 1990) In addition, Wells Fargo Nikko Investment Advisors regarded Ada's ease of maintenance over the long-term (15 year) life span of the software as a significant plus in their selection of a programming language. (Christiansen, 1991)

7. Reliability

Reliability includes the element of higher quality in software delivered and measured in total number of errors. The emphasis on proper software engineering increases the reliability of Ada code.

Strong typing and rigorous compiler validation also reduce program errors. All type checking is enforced at compile time, ensuring that operations conducted are appropriate for the object type. Each implementation requires a database known as the program library, which is an integrated repository containing facts about all compiled units. During compilation, cross-checking is conducted to ensure consistency among units, allowing for early detection of interface mismatch errors.

Rigorous compiler validation procedures (the testing process by which a compiler is verified against the formal definition) also contribute to a decrease in compilation errors.

Several organizations have chosen Ada to achieve maximum software reliability. This is not only because of the language features, but also because Ada encourages and supports good software engineering practices. "Other languages allow or discourage adherence to these principles," according to Motorola. (Brosgol, 1990) Rockwell's

International Space Transportation Systems Division (STSD) cited Ada's reliability, through features such as packages and information hiding, tasking, strong typing and exception handling, when defending their choice of Ada for the space shuttle's backup flight system (BFS). (Crafts, Dec 1989, pp. 9-14)

8. Support for Large, Multi-author Systems

In order to support large and/or multi-author software systems, consistency and coordination between developers and segments must be achieved reliably and efficiently. Ada does this through separate compilation in which all checking is enforced as it would be in a single unit. In addition, it allows a programmer to specify and compile a fixed interface. This ensures adherence to the standards set for the system.

Ada decreases testing and debugging costs, especially for large systems, because of strong typing. Operations on types are verified at compile time, decreasing run-time errors. Also, the modularity of Ada programs and reuse of library packages results in independent and more readily verified components. Ada is similarly cost effective for large systems because of its reliability and maintenance attributes.

GeoMatrix, Ltd., a British company which has implemented a commercial market analysis package in Ada, considered it the most appropriate language for building large, multi-author software systems. (Brosgol, 1990) Thomson-CSF, likewise, believed that Ada provided the greatest support for projects with large teams of programmers (20 or more). The Copenhagen air traffic control system developed by Thomson-CSF was 300,000 lines of code, entirely in Ada. (Brosgol, 1990) One of the largest systems in DoD written in Ada is the Army's Standard Finance System Redesign (STANFINS-R), consisting of over 2.7 million lines of code. It is also a success story for Ada in Management Information Systems (MIS) applications. (Ada IC, 1991)

These are the primary reasons organizations have chosen to implement their software in the Ada programming language. Typically, it is one or two of Ada's advantages which results in its selection. However, it is all of Ada's language advantages which have made it a powerful programming language for all types of applications.

The following chapter will discuss reasons Ada's use is growing as well as factors which are restricting its expansion.

III. ADA ADOPTION AND IMPLEMENTATION

Ada is increasing in popularity as it matures as a programming language. The software industry is changing, and many of the differences have lead to a favorable environment for Ada. However, there still are barriers throughout the community to Ada's adoption.

A. WHY ADA IS INCREASING IN POPULARITY

Changes over the last few years have led to increased use of Ada as a programming language for system development. As a result of the maturing of the Ada environment, the Ada industry has changed. The passing of time has resulted in more knowledge about and experience with Ada by developers and management. The Ada support market, providing compilers and tools, has also grown over the years. Software development has also changed. This is due in part to the rapidly expanding size and complexity of projects. There is also a greater focus on the employment of sound software engineering principles.

1. More Knowledge and Experience

As use of Ada expands, both in defense and non-defense related applications, the documented accomplishments of the language have increased. Its capability and power have gained publicity. Ada advocates are also becoming more vocal. But it is primarily Ada's own technical merits which are receiving the consideration and credit they deserve.

It is generally recognized that one meaningful response to the software crisis lies in the establishment and application of sound software engineering methods. Of paramount importance to sound software engineering are well-trained personnel, integrated tools to support the engineers and the use of good standards like Ada. Ada is a starting point which provides a superior environment for sound software engineering. More organizations recognize this and are embracing Ada as their future language for software development.

The Ada market is no longer subsidized solely by DoD. Corporate and educational participation in Ada's development has expanded significantly in Europe and Asia, as well as in North America. This is evidenced by the number and variety of agencies outside the DoD using Ada. Examples include:

- Government non-DoD (NASA, FAA, Census Bureau)
- Foreign ministries of defense
- Foreign governments (Canada, Japan, Europe)
- Major international corporations (IBM, NTT)
- Colleges and universities worldwide

With the expanding market there has been a significant increase in trained and experienced Ada programmers and developers. Most are trained internally by the organization or through a vendor training program. Penetration into formal educational institutions has also occurred. Ada is being taught at some level in over 200 colleges and universities in the United States alone. It is being taught in every university in England. Ada is also being taught in Poland, China and what was the Soviet Union (although American compiler vendors are prohibited from selling Ada technology to those countries). The growing availability of Ada-trained personnel has made Ada a more attractive option when choosing a language for project development.

2. Better Language Support

Because of the standardization and validation process, the cost and availability of Ada support tools was initially prohibitive. It frequently was enough to discourage project managers from investing in Ada. With the maturation of the Ada industry, validated compilers and integrated software engineering tools supporting Ada have proliferated from a stable, yet developing, array of vendors. The quality and scope of these tools has likewise advanced.

The supply of tools and compilers has grown with the demand. As competition and availability continue to increase, the tendency is for prices to continue to decrease.

Compilers are available for virtually all machines (PCs to Mainframes) and operating systems (DOS to Mac, UNIX to VMS). There are roughly thirty vendors who supply Ada compilers. Costs run from a couple hundred dollars, for a PC compiler, to $400,000, for a multi-user mainframe. Vendors are likewise providing integrated development environments supporting most phases of the software development lifecycle.

Although availability and compile-time speed favor Ada's primary competition (C/C++), corporations are recognizing the value of additional investment in the security and long-term benefits which arise from using Ada. Ada compilers are stringent and more rigidly enforce language standards than other languages. This results in safer and more maintainable code. The generated code also requires less debugging and error correction because of the emphasis on software engineering and strict design prior to programming.

3. Larger, More Complex Projects

As the systems being implemented grow both in size and complexity, the software required to operate and support those systems will likewise grow. Programming and support teams will also grow in number as software projects expand. More personnel will be required for software development and maintenance. This magnifies the significance of integration and documentation.

Ada's strengths in both these areas has been recognized and well publicized. The rigid adherence to standards allows smoother and less time consuming integration of segments of code programmed by different individuals or groups. Sound software engineering stresses the design phase. This improves the cohesion of the system. The readability of Ada enhances the documentation and the maintainability of the code.

The increase in programmer productivity has also contributed to Ada's value in large projects. Decreasing time available and increasing requirements multiply the demands and stress on analysis and development teams. But, organizations which have invested in the greater up front costs required for training and design when using Ada have reaped the long-term benefits of increased developmental efficiency. This in turn has lead to higher productivity and superior quality products.

4. Software Engineering

The growing commitment to software quality, and the decreasing availability of funds to support software development and maintenance, have led to the search for greater effectiveness and efficiency in software development.

In the past, software development has been viewed as an arena of creative individuals working to produce programs as fast as possible. Today, organizations recognize the inherent deleterious effects of this approach. This is especially so when dealing with systems as large and complex as many modern software creations.

Software engineering is a commitment to a quality process in the development of software, and, to a large extent, is independent of an individual's skill. (Riehle, 1991) The focus is placed on the requirements and long-term effects. Quality and productivity have become competitive strategic advantages in the corporate world, propelling the software industry to more fully embrace good software engineering principles.

It is generally recognized that good software engineering is independent of the programming language used. However, Ada offers an environment in which adherence to engineering principles and practices is not only encouraged but enforced within the boundaries of firm Ada standards.

Object-oriented programming (OOP) has, in recent years, generated much interest. The current Ada standard does not provide all the features and generality of OOP. However, Ada's definitional facilities are substantial and are sufficient for software engineering principles such as data abstraction and object-oriented design. Ada 9X (the Ada revision project) is considering enhanced support for OOP.

Recent legislation has mandated Ada's use in all DoD software programs where cost effective. (Schwartz, Dec 10, 1990, p. 1) Recently, there has been a scarcity of waivers to this policy. The following serve as examples of successes of Ada applications developed by DoD:

- **Information systems** - STANFINS-R, Biostatistical Command Summary Reporting System, Tactical Intelligence Manangement System (TIMS), Marine Corps Integrated Material Management System (MIMMS)
- **Communications** - Miniature Receive Terminal, WWMCCS, Net Control Station-JTIDS
- **Tools** - Ada Command Environment, WWMCCS Information System (WIS) Ada Development Tools, Ada Language System/Navy (ALS/N), Common APSE Interface Set (Ada IC, 1991)

Because DoD must use Ada for all applications, the commercial software world is seeing that Ada can be used for non-embedded, non-real-time systems.

Ada's strengths lie in areas which are most useful for real-time, safety-critical applications (error handling, reliability, parallel processing). However, as the non-defense world recognizes the cost savings of well-engineered code, which is more maintainable and reusable, increased usage is expected. Significant improvements oriented to MIS applications in the Ada 9X project, offer greater commercial market utilizations. Ada will serve as a valuable tool in combatting the software crisis.

B. BARRIERS TO ADA ADOPTION

Despite the growing use of Ada as an implementation language, there still exists significant inhibitors, or barriers, to Ada's widespread adoption. These barriers are primarily associated with personnel and management attitudes and biases. Most of the current perceptions of Ada, and its difficulties, were developed early in it's existence, when the language and the development environment were still immature. Major factors inhibiting a more widespread use of Ada are discussed in the following paragraphs.

1. Entrenchment of Other Languages

There is no doubt that Ada is one of the "new kids on the block" as far as programming languages are concerned. Enormous investments in trained personnel and in systems written in C, COBOL and Fortran have restricted entry by most organizations into the Ada market. Developers have been reluctant to pay the initial learning curve, training and investment expenses required to transition to Ada for new software development. At the same time, they would have to maintain personnel and support for existing systems written in other languages. They do not believe that there will be a sufficient return on investment soon enough. Translation or reengineering of existing systems into Ada is likewise ruled out because of a lack of automated tool support for these tasks.

The entrenchment of languages such as C and Pascal runs deeper than the corporate level. These languages are also established in the colleges and universities. This generates a pool of trained personnel entering the market. As individuals are educated in a particular programming language, they become familiar and comfortable with it, shunning conversion to a "foreign" means of programming. The marketplace continues to have a ready and steady source of knowledgeable and trained C, COBOL, et. al. programmers as well as a substantial amount of experience in these "old" programming languages.

2. Complexity

Ada was developed in response to the DoD's requirements and specifications for a language which would replace all other languages. As such, it is a generalized language which has an enormous number of capabilities. One of Ada's strengths is in supporting large, long-lived systems which are subject to changes. When developing smaller, or short-lived systems, programmers frequently prefer the flexibility and speed offered by other programming languages.

The strict enforcement of Ada's standards make it somewhat cumbersome for small applications developed by individuals.

It is also argued that the size of the language and the rigidity of the compiler operation result in code inefficiency and lengthy compilation times. Support for this argument of performance degradation due to language complexity is diminishing as hardware and Ada support tool capabilities grow exponentially.

The attitude towards Ada in some circles is that, because of its size and its power, Ada must be too difficult to learn and use efficiently and effectively. Hence, the argument is that because Ada has too many features, it cannot be used efficiently and cost effectively.

3. Costs

There are several costs associated with the transition from one programming language to another. There are costs of training existing personnel or hiring of trained personnel. New compilers and support tools will be needed. Translation of the old code into the new language would be substantial. If old code is retained until replaced then two languages need to be supported. In that event, old systems would require support and maintenance along with the new systems, duplicating some costs. As with the introduction of any new language, time and productivity losses in the initial stages of transition will also be incurred. This is due to the lack of knowledge and experience of the programmers and developers. Often these costs appear too large to risk (although gains would be achieved in productivity and quality in later endeavors).

The principle of "delayed gratification" is not often a virtue in organizational operations. Cost savings and productivity gains are frequently measured at the outset of project implementation or in the short term.

4. Defense-related, Embedded System Language

The widespread perception is that DoD embedded software systems are significantly over schedule and over budget. Ada, having been developed by DoD, is frequently viewed as a language for defense-related applications alone and is largely to blame for these overages. This is a fallacious perception on the part of the non-defense market. There are, in fact, few specific features for embedded systems incorporated into Ada. Instead, it is a general purpose language which emphasizes early error detection and features to facilitate development and maintenance of large systems.

Roughly fifty percent of Ada applications worldwide are still in defense systems (weapons, command and control, avionics). However, there are a large number of non-defense applications. Significant quantities are noted in space programs (20%) and air traffic control systems (10%). The remaining 20% are found in communications, factory automation, financial and administrative arenas. (Anthes, 1991)

5. Software Development Is An "Art"

The "wild west approach" of the lone programmer is often valued in the United States. The idea remains that too many people spoil the program. Programmer creativity is given priority over product reliability. The stress is on the individual programmer. Give that person whatever is needed to augment personal skill, avoiding "unnecessary" restrictions on creativity, and wait to see the outcome. This approach is fine on small systems with minimal complexity, but, it falls apart when dealing with large systems.

Often, computer programmers are individuals who turned to work on machines because it suits their personalities, requiring little interaction with other people. Software development is seen as an art; it stresses their abilities with problem solving, forming algorithms and relating to their machines. As systems grow, development teams also grow, degrading the programmer's individuality and strengths, forcing them to associate and integrate their work with others.

Ada seemingly strikes a blow at programmer independence and flexibility because of its strict standards which must be adhered to throughout the coding phase. Ada does not affect creativity in system development, which has little to do with the actual language used to implement the system. Ada merely restricts the programmer to guidelines which, when adhered to, generate more reliable, maintainable and portable programs.

The barriers to Ada's adoption and implementation do not occur on a technical level pointing out fatal characteristics of the language itself. The barriers which exist are primarily those of poor perception or misconceptions about the language. They are due to insufficient or distorted information. It is necessary to disseminate accurate information about Ada in order for adequate decisions to be made during the selection of a programming language. Effective implementation of Ada also requires an adjustment in attitudes toward Ada, and management is compelled to take the lead.

DoD executives and senior leadership have embraced the Ada programming language for the entire organization. This adoption by upper level management, along with the legislation requiring Ada's use for all projects has led, and will lead, to Ada's continued growth and strength in the software development field.

This chapter covered many of the positive features and capabilities of Ada, as well as the main barriers which restrict its growth. Ada's continued use in the United States is assured as long as DoD continues to support it. In several foreign countries Ada's growth and use is more substantial. Software development outside the U.S. has changed as a result of the introduction of Ada. The following chapter discusses the use of Ada in other countries.

IV. ADA USE OUTSIDE THE UNITED STATES

A. ADA IN EUROPE

The European Ada industry resembles that in the United States. Ada did not rapidly impact the distribution of programming language market shares. The existence of large, and delicate, software inventories written in other languages, start up costs, as well as the information technology managers' aversion to risk contributed to maintaining the status quo. Market penetration was further hindered by a lack of high-performance Ada tools, low availability of skilled Ada programmers and the perception that only the maintenance phase of the software cycle benefits from the Ada technology.

If success is measured by factors such as the number of programmers, the sales of language oriented products, its use as a teaching language, the portfolio of existing applications, etc., then Ada is not as successful as Pascal or C. Ada continues, however, to benefit from the formal commitment from the U.S. DoD and the fact that NATO has followed suit. This has allowed Ada use to grow as civilian users become increasingly aware of its life cycle cost economies.

1. Ada-Europe

Ada-Europe is an international association whose primary purpose is to promote the use, research, education and knowledge of Ada. Formed in 1988, Ada-Europe is made up of several national associations. They are:

- Ada in Denmark
- Ada Deutschland
- Ada-France
- Ada Greece
- Ada-Ireland
- Ada Nederland
- Ada i Norge
- Ada Scotland
- Ada-Spain
- Ada in Sweden
- Ada Switzerland
- Ada UK

Ada-Europe performs the role of an information clearinghouse on Ada in Europe. It will also represent European industry in the Ada 9X revision process. Ada-Europe publishes a quarterly newsletter (Ada-Europe News) and provides a single contact point for all European Ada users as well as the information dissemination service. (Long, 1991)

The significance of Ada-Europe is increasing as the European community standardizes and consolidates their resources because it represents the Ada community in language related matters.

The number of national associations for Ada users attests to the fact that Ada is a growing force in the software community in Europe. Either as a result of a vocal few or a powerful many, these organizations will continue the spread of the Ada programming language.

2. Military and Government Applications

Military and government use of Ada is extensive. Ada is used by virtually all Western European nations in applications from command and control to embedded weapons systems to communications. Some examples are:

- SACTA - a radar data and flight plan processing system for Spanish Air Traffic Control centers handled by Secelsia
- Tactical combat simulator developed by the Spanish Ministery of Defence
- Multiple weapons control, communications and Ada tool systems for the Swedish Defense Material Administration
- MRCA-TORNADO (Fighter plane) - operational software for the missile control unit developed by the German Ministry of Defense
- EXTRA - library of real-time embedded avionics components sponsored by the French Armament Board/Electronic

- SYTAC - Action and information organization system which is part of the combat system to be installed in a newly designed French ship

- A French government project for a concrete bridge calculation system

- A United Kingdom Atomic Energy Authority project

- Advanced Flight Deck AFDII - software for flight control, flight simulation, engine systems, navigation, atmospheric systems and fuel; sponsored by British Aerospace Commercial Aircraft

- The Copenhagen Airport Traffic control system being developed by Thomson-CSF

- A telediagnostic and telecommand center for Civil Aviation Authorities being developed by Thomson/Syseca

- The Columbus module for the European Space Agency

- DRAGOON - precompiler producing readable and well-structured source code from an object-oriented extension; sponsored by the Commision of European Communities

Because of the relatively smaller size of the European defense and government sectors, the Ada demand in these markets may be lagging behind their counterparts in the United States.

3. Commercial Applications

In some European countries, such as France and Finland, civilian projects outnumber strictly military projects. (KPMG Peat Marwick Consultants, 1988, p. 38) Motivations for choosing Ada range from its aptitude for use with embedded systems to marketing strategy reasons. These were discussed in Chapter II as language advantages. Although control and embedded systems are the predominate systems for which Ada is used, there are increasing applications in management information systems. The variety of nations and applications using Ada is shown in these examples:

- MARC - a Swedish telecommunications system
- A banking application for a Finnish bank being developed by Nokia
- Oil drilling control system managed by Sedco-Forex
- Vigile - an industrial installation supervisor project by CMG
- Recorders for industrial instruments being developed by Enertec
- A PC-based instrument controller for Adret-Schlumberger
- A nuclear plant management system by Cerci
- ISIS-2 robot software for nuclear plants being developed by Hispano-Suiza
- Abeille-Paix project for the insurance sector

Ada was developed by a European team and is represented by a number of prominent proponents in education and industry in Europe. This fact alone may give Ada a head start in European commercial systems development over the American commercial use. However, the European community has made systems development and software technology priorities in its agenda and policies which enhances Ada's opportunity to gain an audience within the market.

B. ADA IN AUSTRALIA

As of early 1991, Ada usage in Australia consisted of 20 defense-related projects, three commercial projects and over ten studies or research and development projects. Several universities in Australia are teaching Ada. Swinburne Institute of Technology and Canberra University both proclaim Ada as the "best language to aid teaching software engineering." (Long, 1991, p. 58)

1. Military Applications

The Australian Defence Forces (ADF) have not mandated the use of Ada, nor do they have a published policy on Ada. (Crafts, Jan. 1991, p. 2) However, Australia is heavily dependent upon American weapons, and because DoD has gone with Ada, Australia has followed. The Australian Department of Defense has sponsored the development of the following projects:

- ANZAC Frigates Program - software for construction of 12 new frigates not necessarily using existing technology
- Australian Navy Submarine Tests Integration - Computer Sciences of Australia will develop the software, integrate and test the system for a new submarine fleet

- Launcher Control System (LCS) - Real-time operation and control of the torpedo tubes on Type 471 submarines under construction (Ada IC, 1991)

The Royal Australian Navy has approached Ada usage with greater enthusiasm than their sister services. The Australian Air Force has had difficulty with the P-3 system which was translated in the U.S. They claim the code was 70% slower as a result. (Crafts, Jan. 1991, p.2)

2. Commercial Applications

Ada faces the same difficulties in Australian industry as in the U.S. There is a general lack of information about Ada technology, or misinformation regarding Ada's features and support. There are, however, some examples of commercial Ada usage:

- Kernal Command Control - an application generator for command, control, communications and intelligence systems
- QUIKTRAK - an automatic vehicle location system
- Remote Control and Maintenance System - test, control and maintenance of airways equipment (radars, etc.)

The Australian software industry is behind the United States in their knowledge of and experience with Ada. The Ada community, however, is equipped with mature Ada language support technology, a software engineering emphasis in the eduction system and a large data base of successful Ada implementations around the world. These will aid the manifestation of Ada in the Australian software development market.

C. ADA IN JAPAN

Although Ada does not support the Japanese language, the emphasis on meticulous engineering and high productivity, accompanied by high quality, has lead Japan to embrace Ada for software development.

1. Military and Government Applications

The Japan Defense Agency has mandated Ada for its embedded systems seeking compatibility with the U.S. DoD and NATO. The Space Development Agency of Japan (NASDA) is also using Ada for new development projects, and is developing its own Ada programming support environment. (Riehle, 1991)

2. Commercial Applications

In the highly competitive electronics industry, the strategic advantage lies within the software. More specifically, quality software. This demand for quality falls right into the domain of Ada.

Likewise, Ada falls right into the Japanese manner of doing business. The Japanese value working as a team, identifying with the success of a project, not personal achievement. Ada provides a software development environment for teams of programmers developing large systems.

The managers of software projects in Japan understand the value of Ada because they have the engineering background. They also appreciate the investment required to receive subsequent rewards.

Ada has been adopted by the Nippon Telegraph and Telephone Corporation (NTT) because of the language's reliability and maintainability. NTT has written more than two million lines of operational code in Ada for communications management software as well as software tools. (Tanaka, 1991) NTT also has an academic division, NED, which owns five small universities teaching courses in engineering. The software engineering curriculum contains an object-oriented design course which uses Ada as the programming language. (Riehle, 1991)

Another corporation which has admitted using Ada is Kawasaki Heavy Industry. However, like many commercial users of Ada in the United States, Japanese commercial users do not publicize their use of Ada because it is treated as a "secret weapon" which is not discussed. (Riehle, 1991)

There are several software development houses in Japan which are developing factory automation, business and software tool systems in Ada. (Riehle, 1991)

Japanese culture embraces the qualities of system development which Ada enhances (diligent engineering, teamwork, productivity, quality, reuse). Consequently, Ada's adoption in Japan appears to have been natural, unencumbered by poor attitudes or misinformed management.

V. THE FUTURE OF SOFTWARE WITH ADA

Ada was developed to remedy the afflictions of DoD's software. The goal was to reduce the high costs of development, maintenance and conversion while increasing system quality. As Ada continues to mature, lowered cost and increased quality will be realized, not only by the defense industry, but, also throughout the software development community. DoD remains a primary participant in the future development and growth of the Ada industry.

A. ISSUES FOR ADA'S FUTURE

Future expectations may be developed by examining past performance. In Ada's case, the future looks bright. Ada performance data exists because some organizations possessed the visionary leadership and courage to take crucial steps toward improvement in the software industry. Ada has demonstrated its viability as an effective tool to overcome software development deficiencies and costs. The following major topics discuss key factors which will impact on the future growth of Ada.

1. The Numbers

There is a debate about whether Ada should remain the required language for DoD software development. Cost/benefit analyses and case studies are stacking up and there is considerable evidence that clearly points toward Ada's advantage over its competition. As the language matures through the Ada 9X revision project and as hardware and support technology advance, it is likely that Ada will remain the language of choice into the 21st Century for DoD.

The Ada and C++ Business Case Analysis conducted by DoD pitted the chosen language of DoD against its most comparable rival. The TRW substudy which examined lifecycle costs concluded that Ada holds a 35% development cost advantage and a 70% maintenance cost advantage. (TRW, 1991) The Software Engineering Institute (SEI) substudy considered technical criteria used by the FAA to evaluate Ada versus C in 1985. In 1991, judging Ada versus C++, SEI also rated Ada higher. (Weiderman, 1991) The report from the Institute for Defense Analysis (IDA) compared the availability of tools and training for each language. IDA gave the advantage to Ada, reporting that there are many more U.S. companies providing validated Ada compilers than C++ compilers, Ada is supported on mainframes while C++ is not, and there is no current standard for C++. (Hook, 1991) The results and conclusions of the Business Case Analysis stated that there is little justification for choosing C++ over Ada.

Organizations which have publicized results from their Ada projects show that long-term costs have decreased 20% over time. These are typically maintenance costs as well as development costs after a full transition to Ada has been accomplished. (Reifer, Nov. 1990, pp. 5-7) Ada results in better designed, higher quality software because of the adherence to software engineering principles and characteristics of the language itself. Higher quality is consistently achieved in the transition period, offsetting the additional costs associated with initial implementation of Ada.

Reifer Consultants discovered from an analysis of over 140 completed Ada projects, consisting of more than 50 million source lines of code, that Ada projects get cheaper as they get larger. Large projects are normally more expensive because of the increased staff size and management required. Although Ada does not reduce the number of staff, it offsets the problems of division of labor and communication because of its structure, methods and tools, along with its reuse potential. Ada's readability softens the learning curve when staff turnover and attrition are considered factors of cost. Reifer's analysis yielded a power law of less than one using COCOMO software estimation. The power law relates size with effort, and Ada's score of 0.95 means an economy of scale exists when a full transition to Ada is made. The findings amplify Ada's productivity advantage over other languages. (Reifer, Jan. 1991, pp. 8-9)

Ada continues to demonstrate that its productivity and cost effectiveness is equal to or greater than other programming languages. What is required from management is a commitment beyond the first few months of implementation and projects written to secure the extensive advantages and future returns from an investment in Ada.

2. Ada 9X

C++ will continue to present the greatest challenge to the Ada mandate because of its support for object-oriented programming and its connection with C. It also continues to mature in the areas of support and standardization. With C++ in high favor, Ada will have to evolve as well to maintain its market. Ada 9X is the significant step in this direction.

The Ada 9X project began in October, 1988 as a revision of ANSI/MIL-STD-1815A to reflect vital requirements. It is intended to be a part of the language maturation and the revised standard is scheduled to be completed in June 1992 for the six month ANSI standardization process. (Long, 1991, pp. 75-79) Guidelines for the development of Ada 9X focus not only on correcting these shortcomings, but also to make the language more usable by programmers and more open to interface with other languages and systems. The project welcomed scrutiny of the Ada 83 language from business, academia, military and all others in order to obtain the most comprehensive review to best enhance the language. Over 750 revision requests were submitted world-wide within one year in response to an invitation to the public. (Ada 9X Requirements document, 1990)

One of the principles pursued in the development of Ada 9X is the minimization of disruption to the Ada community and its infrastructure (tools, education, market, etc.) The least amount of changes possible are to be implemented while solving the recognized problems. The shift should be a natural evolution with the result a seamless integration of the existing Ada 83 capabilities and the enhanced Ada 9X capabilities. (Ada 9X Requirements document, 1990)

A second goal is to avoid any changes which would negatively impact Ada's current advantages as a programming language, such as its readability, maintainability, portability and reusability. Although ambiguities are to be eliminated and the language rules simplified and unified, ease of coding is not to take precedence over support for these advantages.

Some of the specific language enhancements involve the areas of data abstraction, type inheritance, indirect access to data and subprograms, hierarchical library support, more flexible generic parameterization, and others. Additionally, generalizations and simplifications will occur in the areas of discriminants, real arithmetic, attributes, delay statements, text terminators, and others.

Ada 9X will also include functions specifically for business and MIS applications. For example, Ada 9X will accept binary-coded decimal, the format common in many financial applications. It will also support 64-bit fixed point arithmetic which is required for representation of large dollar and cent figures.

3. Reuse

The benefits associated with reuse are obvious (increased productivity, quality and reliability of code). However, substantial costs are associated with reuse (more careful design, library development and maintenance). Other serious impediments to the effectiveness of reuse include business self-interest, lack of incentive and trust. (Syms, 1991)

The potential financial gains of reuse may not be achieved because under most contracts contractors are paid for the lines of code they write. The incentive to use reusable objects is minimal. The questions remain of who will absorb the additional cost of developing reusable code and who will benefit from future cost savings, the contractor or the customer? The government wants to encourage reuse by all software developers to benefit from the cost savings and added reliability, but, the short-term cost benefits do not exist for contractors due to additional design and development costs. Because software and training required for the transition to Ada may cost hundreds of thousands of dollars, and savings from code reuse may not materialize for years, there is little immediate financial incentive for reuse. Programmers also do not have the incentive to use code developed by someone else because it reflects adversely on their productivity numbers.

Operational and managerial changes to handle reuse include proper design of reusable objects and establishment of an infrastructure for reuse (catalog, library facility, organization, etc.). Because initial development of a reuse structure could cost between 10 and 20 percent more than non-reusable code, there needs to be a champion for reuse. Paul Strassmann, the director of defense information, is presently that champion, stating that his "number one [hot] button right now is software reusability." (Menke, 1991) He is willing to invest in the required developmental processes to enhance reuse in the Department of Defense in order to realize the future cost savings. The U.S. Army has opened its Reusable

Ada Products for Information Systems Development (RAPID) Center Library (RCL). It is being used throughout DoD under the Corporate Information Management (CIM) program's data standardization drive.

The key to reuse is sound software engineering, focusing on modular principles as the initial step. The code must then be subjected to rigorous validation and verification procedures. Finally, proper documentation is required to achieve the reusability advantage. Ada encourages and supports modular software engineering and also adheres to strict standards, both in the language and in compiler validation. Ada is a significant vehicle by which copious reuse can be achieved.

4. Perception

The perception of Ada by the software world has been one developed in its initial stages of development - a language created for DoD accompanied by multiple promises, yet incapable of fulfilling its potential. The initial immaturity of the technology, the scarcity and high cost of software development tools and compilers fed the perception that Ada could not be economically applied in solving the software crisis. It seemed that Ada was built to do too much, encumbered by the bureaucracy and cost overruns typically associated with government work. The size and capabilities of

Ada created the impression that it is a programming language which contains too many components and excessive constraints, making it useless, or just more expensive than is warranted. PL/1 is another general purpose language which was to fulfill all programming needs. However, it fell by the wayside because it was too cumbersome to work with. Some feel that Ada will follow PL/1 because of this shared characteristic.

Ada has matured and continues to be refined as a language. The support tools have likewise matured, increasing in complexity and capability as well as availability. With DoD support and industry recognition, Ada will continue to progress as a tool in fighting the software crisis.

Opinions and attitudes often are developed outside the analysis of facts, arising from perceptions which are strongly influenced by our feelings or emotions. The resistance to implementing Ada in DoD ranges from outright defiance to ignorance of the mandate to use Ada. This resistance comes as a reaction to the strict mandate constricting the options of programmers to one language.

Another area from which resistance arises is the requirement for programmers to depart their "comfort zones." They are familiar with C, Fortran, etc. and the unstructured freedom of design with which they have been acquainted. Not only must they learn a new language, they also need to learn a whole method of structured software design. No one likes to be forced into a corner, especially programmers who enjoy developing their wares in a freely creative style. The reactions are more of an attitude, saying: "Who are they to tell me how to write my programs?" Creativity chained and freedom lost are the perceptions arising from this attitude.

Many computer programmers are fond of pointing out that anything which can be done in Ada can be done in another programming language. If this is the case, the converse must also be true. Anything written in another programming language can be written in Ada. Why then this reluctance to accept Ada? Probably fear and discomfort, or perhaps simple laziness.

Management perceptions are as powerful as operator perceptions. Just as no one was ever fired for choosing IBM hardware for a system, in this environment few would be fired for implementing software in C. It is somewhat more difficult to justify the added investment in Ada, especially when quarterly profit statements or annual budget requests reflect increased cost without immediate payback or savings. The understanding that a return on investment will not be immediate, but will be realized in the future, increases the willingness to adopt Ada. Leadership is necessary in this area to see beyond the present and short-term into the long-range planning and return. Investment in Ada is no longer a step of faith, as it was a few years ago. There is extensive experience and continual success which has been documented across the spectrum of applications.

B. RECOMMENDATIONS: WHAT DOD MUST DO

Just like any other aspect of the defense establishment, software development must be planned and programmed to ensure its appropriate role in our security posture. Because Ada is an essential part of defense technology and the future, DoD must maintain an active role in the software development industry and education.

1. Industrial Base

Because the Department of Defense vigorously developed Ada and mandated its use as the standard programming language in the military, Ada made headway into the world of software development in the defense industry, especially with the vast growth of the Defense Department's budget during the 1980s. There was always going to be money and an eventual profit opportunity in the Ada market as long as the U.S. government had a need for national defense.

As the defense budget erodes in the austere nineties, the Ada industry will be faced with the challenge to maintain the technological advancement and evolution without endless government dollars. The Ada industry must meet similar challenges confronting other defense related industries - less investment in research and development as well as procurement. Interest may decrease in Ada technology as the money fades.

The decision to choose Ada as the single programming language for the DoD was an effort to develop a comprehensive software agenda throughout the military in order to decrease costs associated with development and maintenance of multiple languages. By providing a standard language, training and tools were consolidated to simplify the software development, acquisition and maintenance process. DoD must continue to show the same commitment to the quality and decreased lifecycle costs which can be achieved using Ada in a sound software engineering environment. With the current commitment to support and enforce the Ada mandate, DoD has demonstrated a belief and loyalty to the strategy which will bring it through the software crisis while facing a budget crisis.

It is vital that DoD does not submit to use of other high-level languages for specific applications. Specialization resulted in about 450 different programming languages in DoD alone. A firm commitment and strong adherence to the Ada language will convey confidence to the rest of the industry.

DoD must continue an "advertising" program which publicizes Ada's strengths, promoting Ada as the language of choice by organizations outside the government. By consolidating several studies on Ada's capabilities and how they compare with other programming language options, DoD continues to disseminate accurate information about the language, dispelling faulty judgments against Ada. The independent, third party compilation of information about Ada to educate industry may serve to augment Ada's industrial base.

2. Education

The most influential action DoD can take will be in the educational arena. Dynamic support for Ada in the classroom, high school as well as college, will secure Ada's position in the future of computer programming. Since most of the resistance to using Ada comes from those who are unfamiliar with the language, teaching it at all levels of learning will reduce the opposition to Ada by increasing familiarity and knowledge.

A major factor in the success of Ada in the educational system is Ada's ability to enhance the software engineering requirements of future program development. Software engineering is growing as a recognized discipline which must be formally trained. Ada is the best language for instruction in that area.

Ada is being taught at many universities and there is also a growing emphasis on software engineering. However, C and Pascal still rule the classrooms. This is due, in part, to the number of teachers available for these languages, the availability of comprehensive course material and the support and encouragement these languages receive from the corporate world.

DoD is currently the largest champion of Ada. In order to achieve the acknowledgement and acceptance Ada needs to ensure it's future, DoD must take an active role in the proliferation of Ada into civilian educational systems. DoD should emphasize, and perhaps subsidize, sales of Ada compilers to schools. It should also subsidize course development at various universities.

C. SUMMARY

Ada suffers from a lack of understanding about its capabilities by the software development world. The language itself is sufficient for all that DoD originally specified. With the upcoming revisions through the Ada 9X project, Ada will be a very powerful programming language which can be applied to most systems.

The barriers to Ada's adoption are primarily attitudes and beliefs based upon old or incorrect information. This results in organizations clinging to old and incorrect methods of systems development, perhaps to be swept up by the flood of changing requirements.

The use of Ada has expanded around the globe, to all continents, in many varied applications. This use has not been mandated by the government, but instead, is evidence that Ada helps organizations to reduce costs associated with software development and maintenance as well as produce high quality systems.

It has been suggested that "DoD...has the problem of minimizing its own visible connection" to Ada because Ada's link to the military continues to inhibit broader acceptance. (Hext, 1991) However, Cobol originated in the military and does not appear to be suffering any ill effects because of it. DoD will remain the largest user and contractor of Ada-based systems for quite some time and should remain its major adherent and benefactor.

APPENDIX

AUSTRALIA

Universal Defence Systems

Project title: Kernal Command Control (KCC)

Application area: C3I

Number of lines of Ada: >500,000

Address: Suite 3, Enterprise Unit 3
Brodie Hill Drive
W.A. Technology Park
Bentley, Western Australia 6102

The KCC is essentially an Ada application generator designed to provide a basic core of set functions found in all command and control systems. This core is composed of a number of reusable object-oriented Ada packages. Using the pre-tested Ada objects it is possible to meet a wide range of specific application requirements at a fraction of the cost and risk of conventional approaches.

The KCC has also found some remarkable commercial applications, notably the Vessel Tracking System (VTS) and the Irrigation Monitoring System (IMS). The VTS can track either vessels or vehicles, and relies on satellite based technology to obtain positional information which is then imposed onto a map of the locality. In the case of IMS, the incoming data consists of sensor information (eg. moisture sensors) and possibly weather predictions obtained automatically from local weather stations, which is used to generate automatic watering programs.

Universal Defence Systems has always believed in software engineering, which is why it is committed to the use of Ada from the very beginning. Because of this philosophy, UDS did not have to translate code from an obsolete language, with all the inherent design risks. Instead a true object-oriented approach exploited the power of Ada to produce the enormously powerful and flexible KCC.

Advanced Systems Research

Project title: QUIKTRAK Automatic Vehicle Location (AVL) System

Application area: Tracking and location

Target machine: 68000

Man-years for total development: 16

Address: 31 Bridge Street
Pymble NSW 2073, Australia

QUIKTRAK is an automatic location system designed specifically to locate and track objects, such as behicles, is an urban and suburban environment. Any object to be located carries an inexpensive transponder shich, on demand or autonomously, briefly emits a low-power, spread spectrum radio signal. Vehicle tracking information is presented on the customer's PC, superimposed on a background map of the coverage area.

Adacel

Project name: Remote Control and Maintenance System (RCMS)

Application area: Remote anaylysis, test and control

Target machine: PS/2

Man-years for total development: about 10

Address: 875 Glenhuntly Road
Caulfield South
Victoria, Australia 3162

The RCMS has been produced for the Airways Corporation of New Zealand by Thompson Radar Australia Corporation Ltd and the role of Adacel is to develop system requirements, analyze and design the application software, integrate and support the system and provide operator training.

Project features include multiple protocols, real-time fault analysis, graphic user interface and Oracle database backend control. The software is being developed in Ada under DOD-STD-2167 under OS/2.

BELGIUM

BATS S.A.

Project title: Radar Training Simulator

Application area: Military and civilian operator training

Number of lines of Ada: 80,000

Target machine(s): 386

Development host(s): VAX/VMS

Other key tools used: Case TeamWORK/Ada

Man-years for total development: 20

Address: Avenue des Noisetiers
Parc Indusktriel de Recherches du Sart Tilman
B - 4900 ANGLEUR
Belgium

CHILE

Empresa Nacional de Aeronautica (ENAER)

Application area: Avionics

Target machine: Data General AViiON (Motorola 88000)

Address: Rosas 1444, Santiago

ENAER has extensive experience in developing real-time software systems for avionics applications. The company has one of the largest groups of Ada programmers in South America.

After having used C for some time, this is the first use of Ada for real-time applications. The prototyping was done more quickly than it would have been in C.

FRANCE

Thomson-CSF

Application area: Air traffic control

Number of lines of Ada: 300,000

Target machine: Data General MV 10000

Address:
18, avenue de Marechal Juin
92363 Meudon La Foret Cedex

The Air Traffic Control Department of Thomson-CSF's Defence and Control Systems division supplies complete systems for in-flight detection and guidance of aircraft.

Ada was initially introduced in the Copenhagen ATC center in February 1988. Since then, Ada has been used in ATC centers in Kenya and Pakistan, and in simulators in Switzerland and Ireland. Ada is being used in additional ATC projects for Belgium, The Netherlands, New Zealand and Ireland.

Historically, different languages have been used for different stages of the processing (Fortran, Pascal, assembler) resulting in excessive costs. To solve this problem, the Thomson-CSF ATC department chose Ada in 1984 as the common language for all software on the whole processing line. The selection was based on the requirement for software reliability, support for large teams of programmers, ease of maintenance, software reusability and portability, and real-time performance.

CMG

Application area: Turnkey systems for signal processing, industrial automation and computer-assisted production management

Target machine: PC

Address: Z.A. de Courtaboeuf
Rue de la Terre de Feu B.P. 38
91942 Les Ulis cedex

The application was a set of console drivers on a PC integrated into the VIGILE industrial installation supervisor product (process control, alarm management).

Ada was chosen for its enhancement of programmer productivity and the increased quality and performance of products.

CMG decided to continue with Ada for future software development.

Ecole Normale Superieure

Application area: Chemical analysis

Number of lines of Ada: 50,000

Target machine: PC AT; later ported to Sun 3

Address: Departement de Biologie
Laboratoire de Biochimie du Developpement
UA686 - C.N.R.S.
46, rue d'Ulm
75230 Paris Cedex 05

The Ecole Normale Superieure is a graduate school in the mathematical and biological sciences. It has implemented an Ada program to perform chemical analysis of 2-dimensional electrophoresis gels.

An earlier version of the system was written for an IBM 4341 in a combination of Fortran (for long precision arithmetic), PL/1 (to interface with a database), Pascal and APL. The result was non-maintainable and non-portable. In

1986 the E.N.S. team decided to rewrite the entire application in Ada.

Ada was chosen because of its features for data structuring, modularity and reusability. Though compilation time seemed longer than with other languages, debugging was much easier.

Euristic Systems

Project name: CHRONOS

Application area: Expert system for data acquisition

Number of lines of Ada: 40,000

Target machine(s): Apollo Domain workstation network; PC AT;
DEC VAX; UNIX workstations

Address: 3 bis, rue Pierre-Baudry
92140 Clamart

CHRONOS is an expert system product that incorporates temporal reasoning to manage the continuous acquisition of data. It is useful for applications such as the monitoring of data communications networks, cement works operations or distillery processing, since it can manage the most repetitive tasks while supplying the experienced operator with decision making assistance.

Although most expert systems are written in LISP or C, Ada was chosen for several reasons: real-time primitives including language defined tasking; portability across development operating systems and real-time executives; reusability via packages; suitability with object-oriented design methods; facility for developing, managing and maintaining very large applications.

Schlumberger/Enertec

Application area: Recording devices

Target machine: PC AT

Address: 1, rue Nieuport
78141 Velizy Villacoublay

Enertec produces recording devices used in instrumentation systems for applications including aeronautics, satellite reception and control of nuclear power stations.

The Ada application comprised reusable packages for monitoring of instruments and recording devices and a screen interface.

Ada software will be ported to larger configurations (HP 500). Ada is planned for other projects.

TOTAL

Project name: Promethee

Application area: Computer assisted extraction of oil products

Number of lines of Ada: 250,000 - 300,000

Target machine(s): 68020 & industrial I/O boards linked through Ethernet to a HP9000/320

Address: Tour TOTAL
92069 Paris La Defense

The Promethee project involves the implementation of a tool to simulate, in a laboratory, the physical conditions of an oil field. This requires controlling 4000 input/output devices to manage parameters such as pressure, temperature and flow. The system must be capable of running for 4 weeks at a time without stopping and without requiring operator intervention.

Traditionally, TOTAL has used Fortran. Ada was chosen for improved software reliability.

GERMANY

dSPACE Digital Signal Processing

Application area: Fast system control tools

Target machine: PC AT

Address: An der Schonen Aussicht 2
D-4790 Paderborn

dSPACE is devoted to developing and marketing tools to control fast systems such as Winchester disk drives, vehicle suspension systems and robots.

dSPACE is using Ada to implement the software control element of the toolset.

Ada was chosen for language features such as strong type checking, overloading, default parameters, exceptions, packages, data structuring and tasking. Also favored were the ease of maintenance and portability.

Analysis of assembly code and benchmark results show that concerns about Ada's run-time performance are unfounded.

NETHERLANDS

Royal Post and Telecommunications Company (PTT)

Application area: Telephone control and monitoring

Number of lines of Ada: 20,000

Target machine: 286

Man-years for total development: approx 1.5

Address: Bichhorstlaan 36
2516 Bedenhaag
Netherlands

The telephone control and monitoring system provides services for international telephone use, the sending and receiving of faxes and telex messages.

For the control and monitoring of telephone booths and the financial services, the clerks at each counter use a color monitor with touch-sensitive screen controlled by a personal computer. These computers are in turn connected via serial links to specific hardware to control the telephone lines.

The software on each PC performs a number of functions: managing the serial links; monitoring and controlling up to 14 telephone lines; presenting graphical output on the color monitor; assimilating interrupt-driven touchscreen data; presenting menus and controlling selection; handling entries of financial transactions; controlling and buffering for a ticket printer; logging financial and technical data; condensing data for management information purposes.

Ada's high-level tasking was well suited to the application requirements. The extensive compile-time checking reduced the need for debugging generated code. The source code is readable and maintainable. Experienced and sufficiently educated programmers have no difficulty learning the language and exploiting its capabilities.

PTT will use Ada again, for larger projects.

SWEDEN

Bofors Electronics AB

Project title: Ship System 2000 or 9LV MK3

Application area: Complete naval system (C3 & weapon control)

Number of lines of Ada: approx. 1,000,000

Target machine(s): M68020

Development host(s): Rational, VAX

Other key tools used: VAX-based CMS, EXCO

Address: S-175 88 Jarfalla
Sweden

Telelogic

Project title: MARC

Application area: Telecommunications

Number of lines of Ada: 29,000

Target machine: MicroVax

Address: 149 80 Nynashamn
Sweden

MARC is Swedish Telecom's system for supervision of Private Automatic Branch eXchanges (PABXs). The development of MARC started in Fall 1983, making it one of Sweden's first Ada applications. MARC has had several major releases.

MARC is a centralized system that monitors the operation of PABXs and provides: real-time alarm reporting; collection of alarm data, performance data and calls data; a security gateway to the PABX; automatic fault correction; and can handle up to 15 users at the same time and 10 concurrent PABX sessions.

Ada's package facility provided good support for OOD. Ada tasks provided a good match to a number of system requirements. The integration test phase proved to be much shorter in MARC compared with traditional projects. Productivity on MARC 2 was 480 statements per person-month, about 30% higher than an average project.

Volvo

Application area: Area controller computers

Number of lines of Ada: 27,000

Target machine(s): 68010; 386 (to be upgraded to SPARC)

Address: Uddevalla

The area controllers handle the real-time tasks of scheduling, routing and controlling automatically guided vehicle (AGV) traffic through the warehouses and the factory. AGVs fetch requested parts from the warehouse and deliver them to the workers on the factory floor.

The software was written as two major subsystems of 7,000 and 20,000 lines of code. The subsystems were designed to be integrated at the end of the project, which was very easy.

Due to stringent checking of syntax and semantics during compilation, once the source code got through the compiler it often worked the first time.

UNITED KINGDOM

AXL4

Project title: The Implementation of a MASCOT 3 design in Ada

Application area: Naval C3 Systems

Number of lines of Ada: 33,000

Target machine(s): 68030/VAX

Development host(s) (if different): VAX

Other key tools used: MDSE (MASCOT Design Support Environment) & MASCOT 3 Organizer

Address: ARE Portsdown
Portsmouth PO6 4AA
Hants

Brighton Polytechnic

Application Area: Teaching

Target machine(s): VAX, IBM PC

Teaching of second year undergraduate degree program

British Aerospace (Military Aircraft) Ltd.

Project title: ETA

Application area: Military airborne applications

Number of lines of Ada: 60,000

Target machine(s): 68020

Development host(s) (if different): VAX

Other key tools used: CORE/HOOD/LDRA/MDS

Address: Saltgrounds Road,
Brough, North Humberside

C.O.R.A.L.

Project title: DACMAN

Application area: Real-time engine test-bed control system

Target machine(s): PC AT

Address: 274a High Street
Aldershot, Hants
GU12 4LZ

Ada was chosen for its potential reusability and portability. Productivity for real-time software written to product standards was 70 lines per person-day after 9 months.

Cranfield Institute of Technology

Project title: Simulation of a Royal Navy Lynx Helicopter Tactical Data System

Application area: Real-time simulation

Number of lines of Ada: 1500

Target machine(s): IBM PC

Man-years for total development: 6 man-months

Address: Department of Electronic System Design
Bedford MK43 0AL

The requirement for this simulation arose from the need to train aircrew how to operate this new type of equipment. Software engineering techniques were strictly adhered to throughout the project. Requirements analysis was carried out using the actual system's functional specification as a guideline. The project achieved all its aims and objectives.

The project proved the feasibility of using low cost personal computers using Ada for simulations of this type.

Ferranti International Plc, Naval Systems

Project title: Ferranti AP2000 Autopilot Ada Testbed

Application area: Submarine autopilot

Number of lines of Ada: approx. 3000

Target machine(s): 68020

Development host(s): VAX

Man-years for total development: approx. 2/3

Address: Submarine Control Group
North Crawley Road
Newport Pagnell MK16 9HT

This was a feasibility exercise to demonstrate that a novel autopilot algorithm could be implemented in Ada and run in real time on a 68020-based target.

The exercise showed that the implementation was feasible and provided valuable experience of using Ada. The testbed software will be reused as the vehicle for testing of future developments of the algorithm and for a real-time submarine dynamics simulator.

GeoMatrix Ltd.

Project title: ProSpex

Application area: Market analysis

Number of lines in Ada: 60,000

Target machine(s): 286, 386

Address: The Cooper Building
Sheffield Science Park
Arundel Street
Sheffield S1 2NS

The ProSpex market analysis package brings together base maps, database management, statistical analysis and report generation (including map plotting) into a single tool. Users can define their area of marketing interest through postal sector boundaries, post codes, distance or drive time. Alternatively, industry standard territories can be loaded as base maps.

Link-Miles Ltd.

Project title: Ada Transition

Application area: Training simulation

Target machine(s): 386

Development host: Sun 3

Other key tools used: CASE

Address: Churchill Industrial Estate
Lancing
W. Sussex

The Link-Miles Ada Transition program has successfully established a center of excellence in the areas of Object Oriented Design (OOD) and Ada development skills.

A flight simulator major function, digital control loading, has been designed and implemented, using OOD and Ada, as proof of the concept. The results conclusively demonstrated that this approach has no performance disadvantages for real-time embedded software.

The Numerical Algorithms Group Limited

Project title: NAG Ada Library

Application area: Solution of numerical problems

Number of lines of Ada: about 500,000

Target machine(s): VAX, MICROVAX, Sun Workstation

Development host(s): MICROVAX

Man-years for total development: 20

Address: Wilkinson House
Jordan Hill Road
Oxford OX2 8DR

The NAG Ada Library is a comprehensive collection of "algorithms" for the solution of numerical problems. To encourage expandability and reusability, the NAG Ada Library makes extensive use of generic templates. Most of the library units are generic with respect to some number of generic parameters in order to provide greater generality of use as envisaged in the Ada language.

N.A. Software Ltd.

Project title: Numerical libraries

Application area: Numerical analysis

Target machine: Various

Man-years for total development: 6 (to date)

Address: The Merseyside Innovation Centre
131 Mount Pleasant
Liverpool L3 5TF

This range of numeric libraries is being developed for two reasons:

1. The increasing recognition of the benefits of Ada coupled with the mandatory use of Ada by most Western defence ministries would assure a strong future for the language.

2. In recognition that one of the stated aims of Ada is to facilitate a software components industry.

The five libraries being developed are a scalable computer systems library, a statistics library, a fixed-point library, a signal processing library and a variable precision floating-point library.

Orbitel Mobile Communications

Application area: Real-time control of cellular phone tranceivers

Target machine(s): 68020

Development host: VAX/VMS

Address: The Keytech Centre
Ashwood Way
Basingstoke, Hampshire
RG23 8BG

Ada was chosen in order to maximize reliability, maintainability and portability.

PAFEC Ltd.

Project title: HORSES

Application area: Software tools for developing engineering/scientific applications and some end-user engineering applications.

Number of lines of Ada: 200,000

Target machines: Apollo, VAX, SUNsparc

Other key tools used: ORACLE RDBMS

Man-years for total development: 75

Address: 39 Nottingham Road
Stapleford
Nottingham NG9 8AD

PAFEC is Europe's largest independent engineering software house, developing markets and supporting a wide range of integrated software products, primarily for technical computing applications.

Since 1986, PAFEC has been using Ada as its principle programming language. HORSES is PAFEC's software developers' toolkit, consisting of software tools and reusable utility modules to reduce life cycle costs for engineering and scientific applications. Principal tools within the toolkit include user interface management systems, and integrated graphics and database utilities.

Racal Communications Systems Ltd.

Application area: Military command and control

Number of lines of Ada: 62,000

Target machine: 386

Man-years for total development: 19

Address: Western Road
Bracknell
Berks. RG12 1RG

The software architecture is interesting in that it consists of a single Ada program which elaborates differently in each processor node to provide the set of operational facilities required for that node. Flexibility is the key property of the design - different configurations of controlled equipment, applications functions and network elements all have to be supported from a set of reusable software components.

The use of Ada was not a mandatory requirement for this project. It was chosen for commercial reasons because, being based on sound software engineering principles, it was felt likely to result in improved development productivity and better code quality.

Although this was the first Ada project undertaken by the company, which had to bear the introductory costs of acquiring new tools and skills, the investment has been well justified. Significant gains in productivity and reliability have already been achieved and further improvements are expected as Ada expertise and the library of reusable Ada components continue to grow.

Plessey Defence Systems

Application area: Real-time defense systems

Target machine(s): multiple

Address: Grange Road
Christchurch, Dorset
BH 23 4JE

Systems include naval surface ship sonar and an engine monitoring system for the Harrier GR5 and AV-8B aircraft.

Process Plant and Chemicals, Ltd.

Application area: Process control

Number of lines of Ada: 20,000

Target machine: PC AT

Address: 177 Bath Road
Slough, Berkshire, SL1 4AE

PPC is a metal finishing company that supplies equipment to treat manufacturers' parts through a series of open-tank chemical and electrochemical processes. The parts vary widely and range from toys and costume jewelry to automobile engines and military hardware.

The developed system was an application to perform the necessary process control, taking account of process temperatures, solution concentrations, DC electrical supply, treatment times and history of individual parts treatment. The system had to be flexible enough so that users could adapt the control parameters to meet their individual requirements, and it also had to generate efficiency and production reports.

University of Southampton

Project title: Distributed systems

Application area: Ada language and runtime systems

Number of lines of Ada: 2000 (for experimental programs)

Target machine: transputers

Man-years for total development: 5

Address: Dept of Electronics and Computer Science
Southampton S09 5NH UK

Application area: Modeling of deep sea ecosystem

Number of lines of Ada: 2000

Target machine: IBM 3090

Address: Department of Oceanography
Southampton S09 5NH UK

Ada was chosen for this application because of its support for OOD, the language efficiency and its reliability through features such as packages (information hiding), tasks, types and exceptions.

Systematic Software Engineering A/S

Project title: IRIS

Application area: Military communications

Number of lines of Ada: 220,000

Target machine: 8086

Man-years for total development: 16

IRIS is a unique military message formatting tool for entry, editing, formatting and distribution systems, with possibilities for user definition of message structures and composition.

Project title: Alsys Ada MetaGraphic interface

Application area: All

Target machine: 8086

Development host(s): 286

Man-years for total development: 0.5

GRACE is a full-graphic finger-touch based, combined radar monitor/controller and clutter-map display running on PC-DOS. By means of multitasking, GRACE can communicate with 12 radar simultaneously on either synchronous or asynchronous data lines. All radar management orders are executed by finger-touch on graphic push-pull buttions, pop-up calculators and multiple choice selections.

The University College of Wales

Application area: Teaching and research

Target machine: Sun 3

Address: Department of Computer Science
Penglais
Aberystwyth
SY23 3BZ

The undergraduate course at UCW was, and is, firmly oriented towards software engineering and there was growing dissatisfaction with Pascal. In 1986, the change was made to Ada.

The experience with Ada as the main teaching language has been good. Most importantly, student reaction has, on the whole, proved very positive. The quality of the courses has improved significantly as a result of the switch to Ada.

The one serious difficulty experienced is that interfaces between Ada and other system software (such as SunView, X-Windows or the basic operating system calls) either do not exist or are mysterious, undocumented and ridden with bugs.

YARD Ltd.

Project title: Implementation of a MASCOT 3 Design in Ada

Application area: Command and control systems

Number of lines of Ada: 32,000

Target machine: MUME 147

Development host(s): VAX station II

Other key toold used: MDSE, Ready Systems RTTAda, GKS

Man-years for total development: 3

Address: Charing Cross Tower
            Glasgow G2 4PP

The aims of the project are to demonstrate the feasibility of designing and constructing real-time Ada systems using MASCOT 3 and to establish a benchmark against which various target compiler/hardware combinations can be measured.

The system is a simulation of the Close-In-Weapons (CIW) facility of a hypothetical command and control system. Targets detected by the CIW are attacked unless the operator intervenes to veto the attack.

United States

Genesis Software

Project title: Prompt PayMaster (PPM)

Application area: Office automation

Number of lines of Ada: 250,000

Target machine: Wang VS

Man-years for total development: approx. 5

Address: Executive Hills East
10401 Holmes Road, Suite 210
Kansas City, Mo 64131

PPM is a bill-paying system that includes advanced office automation technology such as imaging, voice recognition and networking communications. The system provides automatic computation of balances, due dates and interest, tracking of all payments, electronic fund transfer, audit and payment approval, foreign currency calculations, electronic signature certification, on-line management reports and automatic check generation.

PPM can be ported to another Ada platform in 60-90 days.

LDS Hospital

Project name: Medical Decision Support System

Application area: Decision support

Number of lines of Ada: 40,000

Target machine: 286

Address: 325 Eighth Avenue
Salt Lake City, Ut 84143

The Medical Decision Support System was built for NASA Space Station. It provides data collection and storage; medical record and reference information system; inventory management system for medical supplies and pharmaceuticals; and video, audio and data communications between the medical officer in the Space Station and ground-based medical personnel. The system comprises multiple databases, a query language, an alerting system and a medical dictionary.

SYSCON Corp.

Application area: Access low-level PC hardware

Target machine: PC AT

Address: Suite 210, 9841 Broken Land Parkway
Columbia, Md 21045

The set of packages developed allow an Ada program to access low-level PC hardware features: ROM/BIOS calls, keyboard control, DMA data transfers, interrupt support, mouse input and EGA display control. On top of these low-level packages, SYSCON developed a full text-based windowing system and a pulldown menu system.

Using these packages, SYSCON implemented a real-time software development workstation. The workstation allows the user to use a mouse to manipulate various windows that represent devices from a remote development system. It provides control of a remote embedded system via a simulated front panel interface that allows users to push buttons on the remote computer. The workstation provides a packet protocol for communication with the development system.

LIST OF REFERENCES

1. Ada Information Clearinghouse (IC), *The Ada Usage Database*, IIT Research Institute, 1991.

2. Anthes, G.H., "Ada Making Its Mark at Commercial Sites," *Computerworld*, v. 25, no. 24, pp. 69-71, June 17, 1991.

3. Brosgol, B.M. Ed., *Ada Application Summaries*, Association for Computing Machinery, Inc., 1990.

4. Christiansen, E., "Ada in the Commercial World," Wells Fargo Nikko Investment Advisors , 1991.

5. Christiansen, E., "Investment Analysis," paper presented at the Tri-Ada '91 Conference, 21 -25 October 1991.

6. Crafts, R.E. Ed., "U.S. Marines Using Ada for MIS/Bar Coding Applications," *Ada Strategies*, May 1989, v. 3, no. 5, pp. 1-3, Software Strategies and Tactics, Inc., 1989.

7. Crafts, R.E. Ed., "Motorola Using Ada for Commercial Communications," *Ada Strategies*, July 1989, v. 3, no. 7, pp. 9-12, Software Strategies and Tactics, Inc., 1989.

8. Crafts, R.E. Ed., "Rockwell's Ada Strategy -- A Model for Success," *Ada Strategies*, December 1989, v. 3, no. 12, pp. 9-14, Software Strategies and Tactics, Inc., 1989.

9. Crafts, R.E. Ed., "Major Ada Success: Manufacturing Volvos!," *Ada Strategies*, June 1990, v. 4, no. 6, pp. 3-8, Software Strategies and Tactics, Inc., 1990.

10. Crafts, R.E. Ed., "Case-Ada--An Update on Ada in Australia," *Ada Strategies*, January 1991, v. 5, no. 1, pp. 1-7, Software Strategies and Tactics, Inc., 1991.

11. Focused Ada Research Corp., Market analysis and data base, FAR, Corp., 1991.

12. Hext, J., "Educating Ada: Why, What, How, and When?," *Tri-Ada '91 Conference Proceedings*, Association for Computing Machinery, Inc., 1991.

13. Hook, A.A., et al., "Availability of Ada and C++ Compilers, Tools, Education, and Training," *Ada and C++ Business Case Analysis*, Institute for Defense Analysis, 1991.

14. KPMG Peat Marwick Consultants, "The European Ada Industry: A Study for the Commission of the European Community," *Ada-Related Activities Under the Multi-Annual Programme in the Field of Data Processing*, Commission of the European Communities, 1990.

15. Long, F. ed., *Ada Yearbook 1991*, Chapman & Hall, 1991.

16. Menke, S.M., "Strassmann's Hot Button is Software Reusability," *Government Computer News*, v. 10, no. 18, September 2, 1991, Ziff-Davis Publishing Co., 1991.

17. Office of the Under Secretary of Defense for Acquisition, *Ada 9X Project Report, Ada 9X Requirements*, Carnegie Mellon University, 1990.

18. Reifer, D.J., "The Economics of Ada--Guest Column," *Ada Strategies*, November 1990, v. 4, no. 11, pp. 5-7, Software Strategies and Tactics, Inc., 1990.

19. Reifer, D.J., "Economics of Ada--Guest Column," *Ada Strategies*, January 1991, v.5 no. 1,pp. 8-9, Software Strategies and Tactics, Inc., 1991.

20. Riehle, R., "Ada in Japan," *Embedded Systems Programming*, pp. 28-33, August 1991.

21. Schwartz, K.D., "Ada Use Is Mandatory As Of June," *Government Computer News*, v. 9, no. 26, p. 1, December 10, 1990.

22. Skansholm, J., *Ada From The Beginning*, Addison-Wesley Publishing Co., 1988.

23. Software Strategies and Tactics, Inc., "Anthology of Commercial Ada Applications: Usage and Issues for Non-Weapons Software Systems," SS&T, Inc., 1991.

24. Syms, T. & C.L. Braun, "Software Reuse: Customer vs. Contractor Point-Counterpoint," *Ada: The Choice for '92*, pp. 326-337, Springer-Verlag, 1991.

25. Tanaka, K., "Using Ada at NTT," *Ada Letters*, pp. 92-95, January/February 1991.

26. TRW, Inc., "Ada and C++: A Lifecycle Cost Analysis," *Ada and C++ Business Case Analysis*, TRW, Inc., 1991.

27. United States Department of Defense, *Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A-1983*, 17 February 1983.

28. Weiderman, N.H., "A Comparison of Ada 83 and C++," *Ada and C++ Business Case Analysis*, Software Engineering Institute, 1991.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Commandant of the Marine Corps 1
   Code TE 06
   Headquarters, U.S. Marine Corps
   Washington, D.C. 20380-0001

4. Professor M.J. McCaffrey 1
   Code AS/MF
   Naval Postgraduate School
   Monterey, California 93943-5100

5. Captain Warren J. Soong 2
   8 Braxton Dr.
   Sterling, Virginia 22170

845-216

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CALIFORNIA 93943-5002

Thesis
S665765 Soong
c.1 Ada implementation issues as discovered through a literature survey of applications outside the United States.

AMERICAN BINDERY
2
9-93

3 2768 00018408 9